CLAIMS

What is claimed is:

1. A computer-implemented method in a computer system for interfacing a plurality of software objects, each one of said objects providing at least a first service for at least one other object, each one of said objects requiring at least one service provided by at least one other object, each one of said objects being able to directly invoke code in other objects, and each one of said objects belonging to an object class, said method comprising the steps of:

defining on a first object a plurality of terminals for establishing a connection with another object, said connection providing means for both requesting services of said first object and providing to said first object the service provided by a second object;

assigning to a first terminal on said first object a first persistent identifier that distinguishes said first terminal from all other terminals of said first object, said first persistent identifier identifying the same terminal on all objects of said plurality of objects belonging to a same class as said first object;

assigning a second identifier to a first, yet to be established, connection;

preparing said first object for establishing said first connection on said first terminal in response to provision of said first persistent identifier of said first terminal and said second identifier and preparing a first connection data set sufficient to determine whether said second object can participate in said first connection and sufficient for said second object to invoke code in said first object;

establishing said first connection on said first terminal, in response to provision of said first identifier, said second identifier, and a second connection data set, said second connection data set sufficient to determine whether said first object can participate in said first connection and sufficient for said first object to invoke code in said second object.

2. A computer-implemented method in a computer software system for establishing connections between objects, the system having a plurality of objects, each said object having a plurality of terminals, each said object being able to directly invoke code in other objects, each said terminal and each said connection having an identifier, said method comprising the steps of:

preparing a first object for establishing a first connection on a first terminal by providing the identifier of said first terminal and the identifier of said first connection, and preparing a first connection data set sufficient to determine whether a third object can participate in said first connection, and sufficient for said third object to invoke code in said first object;

preparing a second object for establishing said first connection on a second terminal by providing the identifier of said second terminal and the identifier of said first connection, and preparing a second connection data set sufficient to determine whether a fourth object can participate in said first connection and sufficient for said fourth object to invoke code in said second object;

initiating said first connection on said first terminal of said first object by providing the identifier of said first terminal, the identifier of said first connection and said second connection data set;

determining whether said first object will participate in said first connection based on said second connection data set;

storing in said first object a first portion of said second connection data, said first portion sufficient for said first object to invoke code in said second object;

initiating said first connection on said second terminal of said second object by providing the identifier of said second terminal, the identifier of said first connection and the first connection data set;

determining whether said second object will participate in said first connection based on said first connection data set;

storing in said second object a second portion of said first connection data set, said second portion sufficient for said second object to invoke code in said first object.

3. A computer-implemented method in a computer software system for constructing an assembly object, said assembly object containing a plurality of subordinate objects, each of said objects having a plurality of terminals, each of said objects being able to directly invoke code in other objects, and each of said objects belonging to an object class, said method comprising the steps of:

creating a first subordinate object using a first data set identifying the class of said first subordinate object;

establishing a first connection between a first terminal of said first subordinate object and a second terminal of a second subordinate object using a second data set that identifies said first subordinate object, said first terminal, said second subordinate object and said second terminal;

presenting a third terminal of said first subordinate object as a fourth terminal of said assembly object using a data set that identifies said fourth terminal, said first subordinate object and said third terminal.

4. A computer-implemented method in a computer software system for presenting a terminal of a subordinate object as a terminal of an assembly object, each object belonging to a class, the system having an assembly object, said assembly object having a plurality of subordinate objects, each of said subordinate objects having a plurality of terminals, each of said subordinate objects having an assigned persistent identifier that distinguishes said subordinate object from all other subordinate objects of said assembly object, said method comprising the steps of:

defining on said assembly object a plurality of terminals to establish a plurality of connections with other objects, ach connection providing

both means for requesting services of said assembly object and providing to said assembly object services of other objects;

assigning to a first terminal on said assembly object a persistent identifier that distinguishes said first terminal from all other terminals of said assembly object, and where the same identifier identifies said first terminal on all objects of the same class as said assembly object;

preparing said assembly object for establishing a first connection on said first terminal by providing the identifier of said first terminal and an identifier for said first connection and preparing a first connection data set sufficient to determine whether a second object can participate in said first connection and sufficient for said second object to invoke code in a first subordinate object of said assembly object;

initiating the establishment of said first connection on said first terminal of said assembly object by providing the identifier of said first terminal, the identifier of said first connection and a second connection data set sufficient to determine whether said assembly object can participate in said first connection and sufficient for said assembly object to invoke code in said second object;

obtaining a reference to said first subordinate object and an identifier of a second terminal of said first subordinate object using the identifier of said first terminal and a first identifying data set that identifies said first terminal, said first subordinate object and said second terminal;

delegating an issued first request to said first subordinate object, said first request identifying said first terminal and addressed to said assembly object, by re-issuing said first request to said first subordinate object and substituting the identifier of said first terminal with the identifier of said second terminal.

5. A computer-implemented method in a computer software system for presenting a data property of a subordinat obj ct as a data property of an assembly object in said system, said assembly object having a plurality of

subordinate objects, each of said subordinate objects having a plurality of data properties, each of said data properties identified by a persistent identifier that distinguishes said property from all other properties of the object on which said property is defined, each of said subordinate objects being assigned a persistent identifier that distinguishes said subordinate object from all other subordinate objects of said assembly object, said method comprising the steps of:

providing means to obtain the current value of a first data property of said assembly object and providing the identifier of said first property, and obtaining the current value of said first property;

providing means to modify the current value of said first data property and providing the identifier of said first property and a data value;

obtaining a reference to a first subordinate object and an identifier of a second data property of said first subordinate object using said identifier of said first data property and data that identifies said first data property, said first subordinate object and said second property;

delegating to said first subordinate object an issued first request identifying said first data property and addressed to said assembly object, by re-issuing said first request to said first subordinate object, substituting the identifier of said first data property with the identifier of said second data property.

6. A computer-implemented method in a computer software system for guaranteeing availability of a connection during normal operation of an object, the system having a plurality of objects, each object having a plurality of terminals, a first object requiring that a first connection on a first terminal of said first object be available in order for said first object to conduct its normal operations, said method comprising the steps of:

providing in said first object an inactive state in which said first object does not carry its normal operations and does accept establishing of connections on said first terminal;

providing in said first object an active state in which said first object does conduct its normal operations and does not accept dissolving of connections on said first terminal;

establishing said first connection between said first terminal of said first object and a second terminal of a second object, at a time when said first object is in said inactive state;

requesting that said first object be placed in said active state, where said request will fail if said first connection does not exist at the time of said request,

wherein whenever said first object carries its normal operations, said first connection is guaranteed to exist.

7. A computer-implemented method in a computer software system for establishing a connection between objects with incompatible terminals, the system having a plurality of objects, each said object having a plurality of terminals, where a direct connection between a first terminal of a first object and a second terminal of a second object cannot be established, said method comprising the steps of:

identifying a class of adapter objects, such that:

said adapter objects have a third terminal that can be successfully connected to said first terminal and have a fourth terminal that can be successfully connected to said second terminal,

said adapter objects translate requests that said adapter objects receive through connections on said third terminal into requests that said adapter objects issue through connections on said fourth terminal, and

said adapter objects translate requests that said adapter objects receive through connections on said fourth terminal into requests that said adapter objects issu through connections on said third terminal;

creating an adapter object of said identified class;

establishing a first connection b tween said first terminal of said first object and said third terminal of said adapter object;

establishing a second connection between said second terminal of said second object and said fourth terminal of said adapter object.

8. A computer-implemented method in a computer software system for presenting a plurality of outgoing terminals of subordinate objects as a single terminal of an assembly object containing said subordinate objects, the system having an assembly object and a third object, said assembly object having a first subordinate object and a second subordinate object, said first subordinate object having a first terminal through which said first subordinate object requests services, said second subordinate object having a second terminal through which said second subordinate object requests services, said third object having a third terminal through which said third object provides services, said assembly object having a fourth terminal, said first terminal of said first subordinate object accepting connections with said third terminal of said third object, said second terminal of said second subordinate object accepting connections with said third terminal of said third object, the method comprising the steps of:

preparing for the establishment of a first connection on said fourth terminal of said assembly object by providing the identifier of said fourth terminal and preparing a first connection data set sufficient to determine whether said third object can participate in said first connection;

obtaining a reference to said first subordinate object and the identifier of said first terminal using said identifier of said fourth terminal and a first data that identifies said fourth terminal, said first subordinate object and said first terminal;

obtaining a reference to said second subordinate object and the identifier of said second t rminal, using said identifier of said fourth

terminal and a second data that identifies said fourth terminal, said second subordinate object and said second terminal;

preparing said first subordinate object for a second connection and providing the identifier of said first terminal;

preparing said second subordinate object for a third connection and providing the identifier of said second terminal;

initiating the establishment of said first connection on said fourth terminal of said assembly object by providing the identifier of said fourth terminal and a second connection data set sufficient to determine whether said first subordinate object can participate in said first connection and sufficient for said first subordinate object to invoke code in said third object;

establishing said second connection on said first subordinate object by providing said identifier of said first terminal and said second connection data set;

establishing said third connection on said second subordinate object by providing said identifier of said second terminal and said second connection data set.